BNL 30169

CONF-810759--7

SUMMARY OF THE ACTIVITIES OF THE SUBGROUP ON DATA ACQUISITION AND PROCESSING*

P. L. CONNOLLY	B.N.L.
D. C. DOUGHTY	UNIV. OF PENNSYLVANIA
J. E. ELIAS	FERMILAB
B. GIBBARD	B.N.L.
J. W. HUMPHREY	B.N.L.
P. F. KUNZ	S.L.A.C.
L. LEIPUNER	B.N.L.
W. A. LOVE	B.N.L.
D. MAKOWIECKI	B.N.L.
M. J. MURTAGH	B.N.L.
J. NEIDERER	B.N.L.
E. D. PLATNER	B.N.L.
G. RABINOWITZ	B.N.L.
E. I. ROSENBERG	IOWA STATE UNIV.
M. SCHMIDT	YALE UNIV.
W. SIPPACH	NEVIS LAB.
E. J. SISKIND*	B.N.L.
J. SKELLY	B.N.L.

*Chairman

TABLE OF CONTENTS

*Research carried out under the auspices of the United States Department of Energy under contract no. DE-AC02-76CH00016.

I. INTRODUCTION

A data acquisition and handling subgroup consisting of approximately 20 members met during the 1981 ISABELLE summer study. Discussions were led by members of the BNL ISABELLE Data Acquistion Group (DAG) with lively participation from outside users. Particularly large contributions were made by representatives of BNL experiments 734, 735, and the MPS, as well as the Fermilab Colliding Detector Facility and the SLAC LASS Facility.

In contrast to the 1978 study, the subgroup did not divide its activities into investigations of various individual detectors, but instead attempted to review the current state-of-the-art in the data acquisition, trigger processing, and data handling fields. A series of meetings first reviewed individual pieces of the problem, including status of the Fastbus Project, the Nevis trigger processor, the SLAC 168/E and 3081/E emulators, and efforts within DAG. Additional meetings dealt with the questions involving specifying and building complete data acquisition systems.

For any given problem, a series of possible solutions was proposed by the members of the subgroup. In general, any given solution had both advantages and disadvantages, and there was never any consensus on which approach was best. However, there was agreement that certain problems could only be handled by systems of a given power or greater. What will be given here is a review of various solutions with associated powers, costs, advantages, and disadvantages.

This report makes frequent references to processing capabilities in various units. Table I includes approximate estimates of the capabilities and costs of various standard processors in use in high energy physics.

II. CURRENT HARDWARE TECHNIQUES

We begin by considering digital processing hardware currently in use in the field, and the approximate computing power, cost effectiveness, one-time and per-use research and development costs, and scale of problems associated with each. The numbers quoted are approximate, and are only intended to give a feeling for the ranges involved. We note that in applying a given technique to a range of problems, the costs include a term that is independent of the problem, a term which is proportional to the size of the event under

consideration (e.g. length of data memory), and a term which is proportional to the size of the algorithm employed (e.g. length of program memory).

The units used in the discussion below are machine "instructions" - for reference we note that the complete reduction of a typical event from a large ISABELLE detector is expected to use of order 10^7 of these, and a reasonable trigger algorithm can be expected to use of order a few times 10^4 to 10^5 (e.g. several operations per channel on a 5000 channel calorimeter).

As a point of reference we consider current off-the-shelf medium to large mainframe computers. The cost effectiveness of the processors in such systems is in the range 2-10 instructions per second per dollar. Such machines range in power from around 10^6 instructions per second (IPS) for $\$10^5$ for a VAX and around 1.5×10^6 IPS for $\$2.5\times10^5$ for an IBM 4341 to of order 10^7 IPS for several million dollars for CDC 7600 and IBM 3081 class machines. Typical machines are equipped with an initial memory space sufficiently large to handle any event and algorithm size currently in use (with the exception of older machines such as the CDC 7600 which may require overlaying or use of large core memory to fit large offline production codes). Memory for current machines is approaching the $5000 per megabyte cost range. The advantage of such machines lies in their existence - no R&D is necessary to build a system, and everyone knows how to use them. The disadvantage is in the large scale and correspondingly high price of the peripherals required to support them.

The use of a cluster of commercial microprocessors (1,2) programmed in a high level language such as Fortran has been proposed by a number of groups including both the BNL multi-particle spectrometer and the DAG. Some of the current 16 bit micros (e.g. M68000, Z8000) are not equipped with floating point instruction sets, while others (e.g. 8086/7, LSI-11) have floating point addons implemented either as firmware or additional hardware. Even with these additions, the ratio of floating point to integer execution time is higher than that obtained with typical commercial mainframes. A new generation of floating point chips not designed for any particular processor may correct this, but is not yet generally available. A possible advantage has been the ability of the user to tailor the memory size to the current application. Current scales are single board processors with minimal on-board memory of power in the range $1-2\times10^5$ IPS for $500-$1000, while additional memory runs

$5000 per megabyte. The address space of most micros now runs to at least 4-16 megabytes. The user has the choice of programming in machine language or Fortran, with all of the trade-offs between speed and program development ease that this choice normally entails. In general, the available Fortran compilers do not perform heavy optimization, and while commercial compilers for development systems or cross-compilation do exist, there is a large possibility that the user will have to support an R&D effort of order a man-year to produce a minimal cross compiler to support either the latest hardware release, for which a commercial offering does not yet exist, or a hybrid system containing an integer micro plus floating point from another vendor. However, if a microprocessor were to implement the same fixed and floating point instruction set as an existing "large" system (e.g. LSI-11, rumored LSI-VAX), then the optimizing compiler and debugger already available on such a system could be utilized. The advantages of a system using some number of microprocessors include the possibility of easy programming in a high level language and the relative ease of moving an application which has been developed on an off-the-shelf system to the micros, as well as the fact that the major costs of hardware R&D and debugging have been borne by the chip manufacturer. The disadvantage lies in the number of units it may require to handle large problems, and the corresponding duplication of memory that this would entail. Typical R&D efforts include the development of interfaces to multi-processor busses, cross-compiler generation, and run-time support system creation, each of which is of order a man-year. Once a system is in place, a new program can typically be introduced to the micro system in of order several minutes to an hour.

Emulators of commercial machine instruction sets (3,4,5) are now a fact of life in high energy physics, with more than 50 168/E's in use or under construction. The SLAC LASS group is currently using 4 of them in a single array, and will shortly increase that number to 9, with an estimated effective power of 8.9 processors, in an application involving offline production code. The 168/E has an effective power of 1.5×10^6 IPS at a cost of $3000 plus memory at $15000 per megabyte. The floating point unit has the same performance relative to the integer unit as a typical mainframe, and implements real *4 format with results which are identical bit for bit with those obtained via

IBM execution, as well as real*6 operations which yield results which are statistically indistinguishable from IBM real*8 results for reconstruction codes. The maximum memory size is 128 kilobytes of data and 32 kilowords of program in 24 bit microcode (which corresponds to about 4000 lines of typical Fortran code). Codes which will not fit in this space rquire the use of overlaying from a solid state drum implemented using commercial PDP-11/70 semiconductor memory cards. A run-time system using a PDP-11/04 connected to an IBM channel has been implemented at SLAC. A proposed new version (the 3081/E) will be a more heavily pipelined processor, with a pipeline generated at the time the IBM program image is translated into the microcode executed by the emulator. This effectively makes the machine as efficient in the use of its execution units as an array processor is. The 3081/E will also implement the full IBM real*8 precision, and will address a memory of 16 megawords of program and 2 gigabytes of data. Power is estimated at 3.3×10^6 IPS at a processor cost of $6000. The R&D estimate for the 168/E is 4 man-years for the processor and microcode translator, 3 man-years for the solid state drum and associated "Bermuda Triangle" interface plus software support, and 3 man-years for the PDP-11/04 run-time support system with IBM channel interface and support. Much of this effort will be reused for the 3081/E. The major advantages of such systems are again the ease of programming and debugging using existing optimizing Fortran compilers and their associated run-time library routines, plus the presence of strong floating point hardware support. Once the hardware and run-time support at a particular installation are in place, new applications move into the processor in times of order a day, unless overlay preparation for large codes in the 168/E is required (this occurs when programs exceed the memory addressing capacity, which is .8 that of small core in a 7600). In this case, the current worst case effort required of order a man-month to prepare overlays for an application involving around 10 man-years of programming. The disadvantages of the system include the relative lack of run-time debugging aids for programs already running in the emulator, the requirement of a host processor having the corresponding instruction set on which to develop and compile code, and the possibility of I/O bottlenecks in applications requiring a large ratio of I/O to compute

power (these exist in any processor or processors for which I/O is performed by a support computer with insufficient I/O bandwidth).

Array processors were not discussed very much, possibly because their principal proponents in high energy physics (Wilson et al. (6) of Cornell) were not represented. Their experience is with an AP-190 connected to an IBM system at the computing center. A Fortran compiler ("Aptran") and run-time support system were developed by the group, but programming in machine code is required to take full advantage of the pipeline capabilities of the hardware. When this is done, it is estimated that an average of two of the execution units can be kept occupied in each cycle (in applications involving theoretical calculations for lattice gauge theories). A new hardware release, the AP-164, comes complete with a disk-based operating system and Fortran compiler. Current costs are in the range of 10^5 for machines with 6 MHz clocks (i.e. $6x10^6$ IPS per execution unit) with memory in the $40000 per megabyte range. Advantages of this solution again include the choice of Fortran or machine language programming, while difficulties include the relatively small (but increasing) size of program memories and the apparent difficulty of debugging code, especially that programmed in machine language. With the appearance of the AP-164, R&D costs are decreasing to the scale of off-the-shelf systems.

Another class of processor which received minimal discussion because of lack of proponents was the custom microprogrammed processor (7-10). Typical examples of this type of machine are the FNAL M7 and the CERN ESOP. These can range from simple bit-slice processors to fully pipelined machines with multiple execution units and microarchitectures optimized for specific calculations (e.g. the M7 is optimized to perform one 2-vector dot product per instruction). Typical examples of such machines now are equipped with meta-assemblers for the microcode, but rather minimal debugging tools. Typical ECL machines have a power of a few times 10^7 to 10^8 IPS for a price of order a few times 10^4, while corresponding TTL machines may have both speed and price smaller by a factor of 4-10. Typical program memories contain several thousand machine instructions, and since (to our knowledge) no efforts have been made to overlay memory, large offline production codes have not been executed in this fashion. R&D efforts are typically of order one to a few

man-years to produce the first machine, while programming times of several man-months are required to implement small trigger algorithms (several hundred lines of program). Advantages of this approach center around the relatively large quantum of computing power available in a single piece of hardware and the cost effectiveness of the solution, while disadvantages include the difficulty of writing and debugging code required to achieve this effectiveness and power, and the relative scarcity of people within an experiment with the capability to understand and change the code.

The final class of trigger system discussed was the modular trigger processor under development at Nevis Labs (11), and in use in a less ambitious guise at FNAL (12). This processor design is motivated by noting that any given algorithm can be recast from the usual description as a sequential instruction stream, into a number of operations performed on a data stream (this is what computer scientists call a "data driven computer"). Such an engine is constructed by cabling together a series of basic execution modules each of which is capable of performing only a single type of operation. The programming of such a machine lies in the selection of the identities of the execution modules and in their interconnections. A prototype of such a Nevis machine will shortly be installed in a dimuon experiment at FNAL - a more ambitious effort is part of an experiment scheduled to run at the AGS in 1982 and at FNAL sometime thereafter. An unusual characteristic of this type of processor is that there is no natural scale to such a machine. In particular, if an algorithm requires the summing of an array of numbers, it can be implemented in one execution unit in one unit of time, two execution units in one-half unit of time, and so on. In fact, the designer is free to trade number of execution units for execution time up to a limit which is determined by the log of the number of channels (i.e. the number of stages required to combine the results of independent execution units goes as the log of the number of such units). Although it is certainly not true that one line of Fortran in a conventional processor necessitates a corresponding execution unit in the algorithm, it is generally conceded that it would be difficult to program large offline production codes in such an environment. In particular, the approach seems to require careful thought about a problem in order to reduce it to an elegant and somewhat symmetric formulation (e.g. although

there certainly is a mechanism for making cuts, the consideration of all straight line tracks between hits in two chambers except those that pass through some oddly shaped piece of material between them, which could be handled trivially although not elegantly in conventional Fortran, would seem to involve the proliferation of special purpose and therefore rarely used units). Current costs are of order a few hundred dollars for execution units which can produce 4×10^7 results per second if kept fully occupied. Research and development efforts are estimated at 3-6 man-years to produce an initial system, including cable formats and standard board layouts, as well as simulation tools, while program effort is estimated at one to several man-years to implement a trigger algorithm for a large experiment. These estimates are hard to make at this point, as expertise in this technique resides in a very few individuals. Advantages of the system are the possibility of handling very large data rates in machines containing many execution units and the cost effectiveness of such a processor if the algorithm is constructed to make maximum use of each execution unit, while disadvantages stem from the lack of programming experience, and the programming difficulty encountered in casting algorithms into a form which makes such use of the execution unit, as well as the inability of the approach to tolerate "large" algorithms which contain one-of-a-kind non-symmetries.

III. DIGITAL TRIGGER PROCESSING

We now proceed to a discussion of the applicability of these hardware techniques to trigger processing. None should be thought of as replacements for conventional fast logic triggers, or for analog triggering schemes involving clever energy sums or comparators, etc. For discussion purposes we lump the analog and fast logic portion of the solution into "the pretrigger" and then proceed with calculations to determine what pretrigger rate a digital trigger processor can accommodate.

The implementation of a trigger algorithm in Fortran was estimated at of order 10^4 to 10^5 machine instructions per event. A particular calorimeter trigger suggested by the workshop organizers and involving computing angular moments about jet axes in a calorimeter of 5000 channels was analyzed and estimated at 3×10^4 machine operations. Actual performance of one-half of the

algorithm was measured at 2.25 milliseconds on the CDC 7600, and thus corresponds to 2.25×10^4 operations for this half. For a typical scale we estimate 10^5 operations on a data stream involving 1 KHz input rate from the pretrigger. This involves 10^8 IPS of performance requirement, or a facility of order 10 7600 equivalents. The processing power requirement is of course independent of the actual rejection power achieved by the algorithm.

The use of a microprocessor for a simple programmable trigger algorithm involves the need for a memory of order 100 kilobytes in order to handle a single complete event (estimated to be of order 40 kilobytes) as well as associated program and processed data. This involves a memory of order $500, which is on the same scale as the processor cost. If the algorithm can be implemented without floating point operations, then the micro is not burdened with unnecessary hardware. If not, it might be at a disadvantage. Again, the possibility of programming in both machine code and Fortran, or possibly a mix of both, is attractive. Using current machines of power of order 10^5 IPS would require the management of an array of order 10^3 processor to achieve large scale performance. Careful use of machine code programming and lookup tables could reduce this number substantially.

The use of an existing emulator such as the 168/E would typically not involve the use of an overlaying scheme and associated overheads. The proposed 3081/E is so carefully optimized for floating point performance in a pipeline that it might prove optimal to build a stream data converter from integer to floating point format to take advantage of the increased performance. As yet, there exist no programming tools to program this machine in microcode, but given the optimization performed implicitly in the pipeline generation during microcode generation, the advantage of this is not so apparent. The unit quantum of performance is such that management of an array of perhaps 30 processors of 3081/E scale would be necessary to achieve 10^8 IPS of performance.

Array processor memory size is such that a trigger algorithm and data should fit comfortably. Again, the use of a steam data converter might be necessary. The quantum size is such that large data rates might be processed with of order 10 processors programmed in machine code. The possible use of

a mix of Fortran and machine coded programming might again be advantageous, but one would pay a penalty for lack of Fortran optimization.

The use of custom microcoded processors and the Sippach et al. data driven modular processor in trigger processing was the reason for the development of these techniques. The current scales of these machines are optimized for the types of problems that trigger processing presents, and in the case of machines like the M7, so are the internal microarchitectures. The advantages and disadvantages of these processors in triggering are thus as described in the preceeding section.

In summary, trigger processng by programmable filters seems to have a number of roughly comparable alternatives using commercial microprocessors, emulators, or array processors. Given numbers like 10^4 to 10^5 instructions per event and a capital budget of $\$10^5$ to $\$10^6$, it seems likely that such a filter could accept input events at a rate something like 10^2 to 10^3 Hz. The best hope for a general purpose digital filter to handle input trigger rates orders of magnitude higher than this is the modular data driven trigger processor. Using such a device containing a few hundred to a thousand execution units, it does not appear impossible to handle pretrigger rates in the 10^4 to 10^5 Hz range.

IV. INTERACTION REGION COMPUTING SYSTEMS

A standard picture of an interaction region computer system was proposed by DAG and generally accepted. This system consisted of a real-time data acquisition arm, an interactive computing system body, and a CPU enhancer brain. The physicist at the interaction region basically sees the interactive system, and through that controls and monitors the entire system. The data acquisition system performs essentially all of the real-time computing, including digital filtering and tape writing. This system is capable of furnishing data streams at any level (e.g., both before and after the programmable filter) to the interactive system, and is controlled by the interactive system. Program development for the data acquisition occurs in the interactive host.

A desirable addition to the data acquisition arm is referred to by the FNAL collider group as a 'Human Interface Computer' (HIC). This is an

interactive computer imbedded in the acquisition system in such a way that during the development of the detector, all data in a detector sub-system can be routed to the HIC, and that during operation of the integrated detector the HIC can still 'spy' on the data stream. This concept basically exists in modern large detectors (e.g., UA1, UA2, and several PEP detectors), but the degree of transparency of operation between the setup and operation phases, as well as the human interface (i.e., command language and graphics facilities on the HIC as compared to that on the main interactive computer) vary widely. It is hoped that program development for the HIC would occur in the main interactive computer once the detector subsystem for which a HIC is responsible is integrated into the whole detector.

The interactive computer is responsible for code development for all computers in the interaction region, as well as database management for the entire detector. It is also responsible for directing, but not performing, whatever online analysis is necessary to ensure continued calibration of the detector.

The actual online analysis is performed in the CPU enhancer. This analysis consists of full reconstruction of a few percent of all acquired events in order to calculate efficiencies and resolutions of detector elements, as well as to generate new calibration constants for placement in the data tape and in the data acquisition database. The methods currently available for building such an enhancer are essentially the cluster of microprocessors, emulators, or array processors described above.

It is assumed that the interaction region computing system will be built out of modular components in order to tailor the size of the system at any interaction region to the requirements of the experiment installed there. Therefore, we can now make only rough estimates of the approximate sizes of components. In the previous section, the computing power of the programmable filter was estimated at 10^3 Hz of events requiring 10^4 to 10^5 instructions of processing, for a total of 10^7 to 10^8 IPS power. The size of the interactive system is estimated by assuming that during periods of intense system debugging, experiment construction, and machine down periods, approximately 20-25 percent of the physicists on the experiment will be logged on. This leads us to suggest that the interactive system for a small ISABELLE

experiment be capable of supporting around 10 interactive users, while that for a large experiment be capable of supporting around 20 users. Finally, the CPU enhancer should be capable of reconstructing a few percent of a data stream of 10^1 Hz events, each requiring of the order 10^7 instructions of processing. This corresponds to 10^6 to 10^7 IPS of total power.

The question of whether the architecture of the intersection region computing system, especially its interactive and CPU enhancer sections, should be basically IBM compatible or DEC compatible was discussed at some length and with great spirit. The conversation was limited to these two alternatives only because 'Experts' on these architectures were present, and there was no intent to suggest that these were in fact the only possibilities. At one point, a list of about 30 criteria by which such a selection should be made was displayed, with each side feeling that the criteria in which its offering was superior were the more pressing ones. Although no conclusion of which was a superior architecture was reached (none was expected), this in itself marks quite a change in the state of affairs, since in the past high energy physics online computing has been the province of DEC and similar minicomputer manufacturers. The absence of IBM had been due to high initial entry cost, poor cost/performance ratio, lack of interactive support, and difficulty in interfacing to user equipment for good real-time performance. The scale of ISABELLE experiments has probably now exceeded the entry cost barrier, while the separation of real-time and interactive functions has essentially reduced the user interfacing requirements to a single port to the experiment (presumably Fasthus) without tremendous bandwidth requirements. The advent of the 4341 class processors has brought the cost/performance ratio to within at least a factor of two of the minicomputer manufacturers, with the possibility of future improvements due to competition with Japanese plug-compatible manufacturers. Finally, the appearance of the interactive features of the VM/CMS interactive operating/monitor system means that a small but growing group of high energy physicists (currently primarily at SLAC and several east coast universities) will have experience with IBM interactive computing.

In summary, the subgroup concluded that the interaction region computing system should most likely be centered around an interactive computing system which for the larger ISABELLE experiments should be around twice the size of

the current VAX-11/780, and that this be supplemented by a real-time data acquisition system interfaced to the interactive system at a single point, and by a CPU enhancer at the approximate level of a CDC 7600 for online analysis. It was pointed out that the same techniques used for creating a programmable filter for the data acquisition system of roughly 10 times this power could be used for such a enhancer but there were members of the subgroup who felt that compatibility for use as a CPU enhancer should not necessarily be forced on the design of the filter. Finally, it was realized that with the introduction of the model 4341 processor, as IBM based solution should be seriously considered.

V. DATA PROCESSING

The subgroup considered the problems of data processing, although this was not specifically in its charge, because it was felt to be an associated problem which was not under consideration elsewhere at the summer study, and because it arose naturally out of the discussion of technique. The subgroup was concerned with the scale of both the data storage capacity and the processing capacity required to support ISABELLE.

The assumption in these discussion was that some clever triggering scheme had taken the raw interaction rate of 1-5 MHz at phase I or 50 MHz at phase II, reduced this to a pretrigger rate in the 10^4 to 10^5 Hz range, pushed this through a modular trigger processor to achieve something like two orders of magnitude reduction, and finally through a programmable filter to achieve a rate of 10 Hz. We therefore assume a 10 Hz event rate or 400 kilobyte/second data rate. A further assumption is that a good year of ISABELLE running produces in the range of 1-3 months of actual data recording, and thus 3-10×10^7 events and $1-3\times10^{12}$ bytes of data.

The worry about data storage is where to store 1000-3000 gigabytes of data per experiment per year. A 6250 BPI magtape holds 10^8 bytes, and so this is $1-3\times10^4$ tapes (i.e., a good sized tape library). Actually, this is not such an insoluble problem as long as it isn't allowed to propagate. What needs to be pointed out is that the first stage data reduction had best be a true reduction in volume, and not an expansion in volume. As an interesting comparison, it is widely assumed by the paranoids among us that the federal

government (in the guise of either the IRS or the Census Bureau, probably the former) has possesion of a database describing the citizenry of this country. A reasonable guess would be that a few kilobytes of data per person would suffice. The size of such a database is thus of order 10^{12} bytes or 10^4 6250 BPI magtapes, i.e. around one running month of data from a large ISABELLE detector. The subgroup noted the differentiation between the actual volume of data and the physical volume and number of media required to store the data. The latter problem is subject to future technology developments such as optical disks, while the former is not.

The question of how much processing capacity is required to handle ISABELLE data was discussed at some length. It was suggested that it is useful to split this capacity into a portion proportional to the number of physicists and a portion proportional to the number of events. The first portion is essentially interactive computing required to develop and debug code and test algorithms, as well as to analyze data summary tapes. The second portion is production computing used to create summary tapes and to generate and analyze monte carlo events.

There was general agreement that the amount of interactive and support computing required was in the neighborhood of .1-.2 VAX equivalents (or .01-.02 7600 equivalents) per physicist. This comes from observation of the number of people supported both at HEP computing centers at universities (typically a VAX or IBM 4341) and at accelerator centers (the FNAL CYBER system and the SLAC 3081). Much of this computing power can actually be provided by the home institutions of the ISABELLE users. However, BNL should be prepared to provide interactive computing for BNL participants on ISABELLE experiments, and for graduate students and postdocs from outside institutions in residence at ISABELLE. An estimate of the interactive power required is thus 100-200 physicists in residence at .01-.02 7600's each, or 1-4 7600's worth of interactive computing. This is essentially a long-winded way of saying the obvious, namely that BNL needs an interactive center on the scale of FNAL or SLAC.

The amount of production capacity required is estimated at around one 7600 second per event for reconstruction. At the data rates mentioned above, this corresponds to about 5 7600's per large experiment running all year long

to process the data acquired during that year. These estimates, which are based on current reconstruction figures from R807 at the ISR and from plans for the FNAL collider, and are matched quite well with current rumors about plans for CESR-II, especially when one folds in the possibility of generating and reconstructing one monte carlo event per data event acquired. Kunz of LASS pointed out that these are not especially large numbers, given the fact that SLAC Group B will have a 7600 worth of production capacity in 168/E's by the end of the summer, with plans to expand at a later date.

The DAG and the FNAL collider representative went to great lengths to point out that processing power in a programmable trigger filter was of the right size to address this problem during the period that no data is being taken. In particular, a trigger filter capable of reducing the data rate from 1 kHz to 10 Hz by running events through an algorithm which requires 1 percent of the processing power of the full offline could also run the full offline on the 10 Hz data rate. Certainly, if ISABELLE were to run 4 months of the year, the other 8 months use of the trigger filter could be turned over to production work, provided that sufficient resources existed to manage such a use, and an operator were provided to mount tapes. This would be true regardless of whether the filter were realized in commercial microprocessors, emulators, or array processors. The subgroup agreed that this was a possibility, but refused to come to a consensus on the advisability of such a procedure, and again, as in the case of the use of the programmable filter as the CPU enhancer for online analysis, did not wish to constrain the design of the filter so that it would be useable as an offline engine.

In summary, the subgroup concluded that a 10 Hz trigger rate during several months of the year required the use of the equivalent of tens of thousands of 6250 BPI magnetic tapes for storage of a year's data, and certainly more than 10 7600 equivalents of production capacity to reduce the data (probably implemented in special purpose hardware rather than off the shelf systems), as well as an interactive facility at the level of a few 7600 equivalents to support 100-200 physicists in residence. If data storage and production capacity of this scale are not available at BNL, the only rational alternative would be to further increase the selectivity of the trigger. In the minds of some subgroup members, this corresponds to the progression from

a standard bubble chamber experiment, in which all events are available for subsequent analysis, to a triggered bubble chamber, an option not to be explored until after an initial survey experiment. To others, it was not clear why the reduction in rate from 10 Hz should be considered as "triggering," while that from 1-50 MHz to 10 Hz was considered "standard." However, regardless of trigger rate, it is important to understand that the need for interactive computing remains large.

VI. STANDARDS

The subgroup discussed standards, especially the Fastbus, on several occasions. The majority opinion was that standards are necessary, but only where they are useful.

The question of where to standardize in high energy physics data acquisition is a purely fiscal and political one. As a potential user of a standard, one uses it if it costs less to use an existing standard (or to modify it to one's use) than to construct something equal or better. As a potential inventor of a standard, one invests the time and effort only if one believes that in the end one will make back one's investment.

In high energy physics data acquistion a representative standard product is PDP-11/CAMAC MULTI. FNAL required around 3-5 man-years to take an existing program (which itself took several man-years to develop) and turn it into a standard. Every time a new experiment uses MULTI, that experiment saves itself at least 2 man-years of development. The $100K or so that this represents far outweighs the $10K or so required to purchase the requisite CAMAC hardware, and the time delay is hardware delivery time, not uncertain software production time.

It was assumed that ISABELLE experiments are so big that real standards must exist within each experiment. It is no harder to generate a standard for one experiment than for 6, and thus it pays to standardize for all of ISABELLE. A rough estimate from FNAL MULTI, which is borne out in the ISABELLE controls system, is that half of the actual labor involved in producing such a system is independent of the experiment being performed or the device being controlled. A further estimate is that a complete data acquisition system for a large ISABELLE detector, including data collection,

trigger processing, human interface computing (HIC), downline loading, calibration and online monitoring hardware and software, will require around 50 man-years to produce. The potential benefits from a standard experiment independent ISABELLE data acquisition system will thus run to around 25 man-years for each experiment past the first to use it.

The subgroup considered some of the characteristics that such a system should contain. These included a standard hardware interface (assumably the Fastbus), a standard way to obtain a data buffer containing an event in an interactive program running either on a HIC or on the interaction region interactive computer, a standard way to talk to the console and associated display devices, and a standard command language to use at any console.

The emerging standard interface between high energy physics hardware and computers is the Fastbus, under which there are currently two systems being implemented. The trigroup version is in the advanced design stage, and is anticipated as the standard for the FNAL colliding detector facility. A single crate implementation of BNL-Yale version has been in use for a year at AGS E-735, with a multiple crate version under construction for use in an experiment this winter. If a standard is to be adopted at ISABELLE, it will be needed at an early stage so that detectors being planned can utilize it for tests and debugging. If no standard is adopted, each experiment will develop its own system, and in the long run there will be much wasted effort in such uninteresting but time consuming areas such as backplanes, power supplies, cooling and ventilation, and so on.

VII. RESEARCH AND DEVELOPMENT

The ISABELLE data acquisition group described its current efforts to the subgroup. These included: (1) investigation of current commercial network products for file transfer; (2) development of software to manage multiple microprocessors or emulators for use in programmable trigger filters or CPU enhancers for online processing; (3) possible development of an emulator for the VAX architecture; (4) design of a Fastbus host interface for the VAX-11/780 (with the PEP Mark II and TPC experiments); and (5) participation in the design of the data acquistion system for the FNAL colliding detector

facility in order to gain experience with systems on the scale of those required for ISABELLE detectors.

The subgroup suggested the addition of the following topics: (6) investigation of the control of experiments (e.g., high voltages, gas systems, etc.), possibly through use of standard modules developed for the ISABELLE controls system; and (7) development of guidelines for data acquisition system integration, i.e. providing means such that systems software developed for use during the construction and debugging phases of detector development could be re-used during the actual running of the detector.

In addition, the subgroup expressed its desire for both the standard data acquisition system described in the preceeding section, and for a standard modular data driven trigger processor, provided that upcoming experiments at the AGS and FNAL validate the basic principles of that design. The subgroup did not however, explicitly commend these projects to the attention of the DAG.

VIII. SUMMARY

The subgroup on data acquisition and processing considered a number of computing-related topics during a series of very lively and not particularly highly directed meetings. The major topics and "conclusions" are listed here.

Digital Trigger Processing: A number of possible approaches exist to the problem of providing standard Fortran programmable trigger filters at the level of 10^4 to 10^5 machine instructions on several hundred Hz to 1 KHz event rate. A most promising hardware technique to extend our capabilities beyond this level is the modular data driven trigger processor under development by Sippach et al. at Nevis Labs. It is important to recognize that the triggering problem for an experiment which is to run somewhat continously for several years is solved not when the event rate is reduced to one at which data can be stored for later offline analysis, but rather when that rate is such that the existing analysis capacity can process the data at a rate equal to that at which it accumulates.

Interaction Region Computing Systems: The computing configuration should be built from an interactive host, a real-time data acquisition system, and a CPU enhancer. At a large experiment, the host should be capable of supporting

around 20 users, while the CPU enhancer should have processing power in the range of CDC 7600, but be built out of more cost effective hardware. With the advent of the 4341 class processors, and the VM/CMS interactive operating/monitor system, the use of an IBM compatible architecture merits serious consideration.

Data Processing: The volume of data written on tape or other mass storage is of order 1000 gigabytes, which corresponds to 10,000 6250 BPI tapes (but will hopefully be in more dense media), per large experiment per year at a 10 Hz logging rate. Efforts should be made to reduce the voulme of this data (in the literal sense) at the earliest opportunity as such a data rate corresponds to 6250 BPI tape drive turning at approximately full speed. For such an accumulation rate, the amount of production capacity required to create data summary tapes and generate and reconstruct monte carlo events is at least of the order of 5 CDC 7600's per experiment. This is sufficient to process the data from a year's running during that year. Although it is unlikely that computing on such a scale can be provided with off-the-shelf commercial systems, we believe that there exist techniques which promise an order of magnitude improvement over such a solution. However, regardless of this capacity, experiments should strive to reduce ther trigger rates to match the then available processing capability. In addition to this production capacity, an interactive computing facility of order 1-2 CDC 7600's is required at BNL per 100 physicists in residence at ISABELLE in order to support code generation and debugging as well as physics results preparation by analysis of data summary tapes.

Standard Data Acquistion and the Fastbus: The construction of a standard data acquisition system based on the Fastbus is likely to require of order 50 man-years of hardware and software labor, approximately half of which is experiment independent. The effort required to produce the experiment independent portion as a standard product would be well justified, but will be wasted unless the system will obviously (to the satisfication of experiments) be available for use in time for the first experiment to use it during its debugging phase.

REFERENCES

1. H. Bernstein et al., Preprint BNL 29509, (1981)
2. L.O. Hertzberger, Preprint NIKHEF-H/81-3, (1981)
3. P.F. Kunz et al., IEEE Trans. Nucl. Sc. NS-27, 582, (1980)
4. C. Halatsis et al., Preprint CERN DD/79/7, (1979)
5. R. Grishman, Preprint New York Univ. Puma System, (1978)
6. K.G. Wilson, Preprint Cornell CLNS-81/477, (1981)
7. T.F. Droege et al., IEEE Trans. Nucl. Sc. NS-25, 692, (1978)
8. T. Lingjaerde, Preprint CERN/DD/75-17, (1975)
9. G. Franke, Preprint DESY 80/109, (1980)
10. P. Waloschek, Preprint DESY 80/114, (1980)
11. W. Sippach et al., IEEE Trans. Nucl. Sc. NS-27, 578, (1980)
12. E. Barsotti et al., Preprint Fermilab FN-312, (1978)

TABLE I

System	Processing Power (MIPS)	Processor Cost ($1000)	Year of Introduction
CDC 7600	10	4000	1971
CDC CYBER 175	5	2500	1978
IBM 3081	15	4000	1981
IBM 370/168	3.3	2000	1972
IBM 4341	1.5	250	1978
SLAC 168/E	1.5	10 (Parts only)	1980
DEC VAX-11/780	1.0	100	1977

Summary of processing power and CPU cost for systems in common use for high energy physics data reduction (production). All numbers are approximate to within a factor of two. Costs are for central processor and memory only, and are not intended to indicate cost for a complete computer center based on the indicated processor. Processing power is in "million instructions per second," and is intended to indicate relative performance for a typical floating point instruction content in a system used primarily for production.